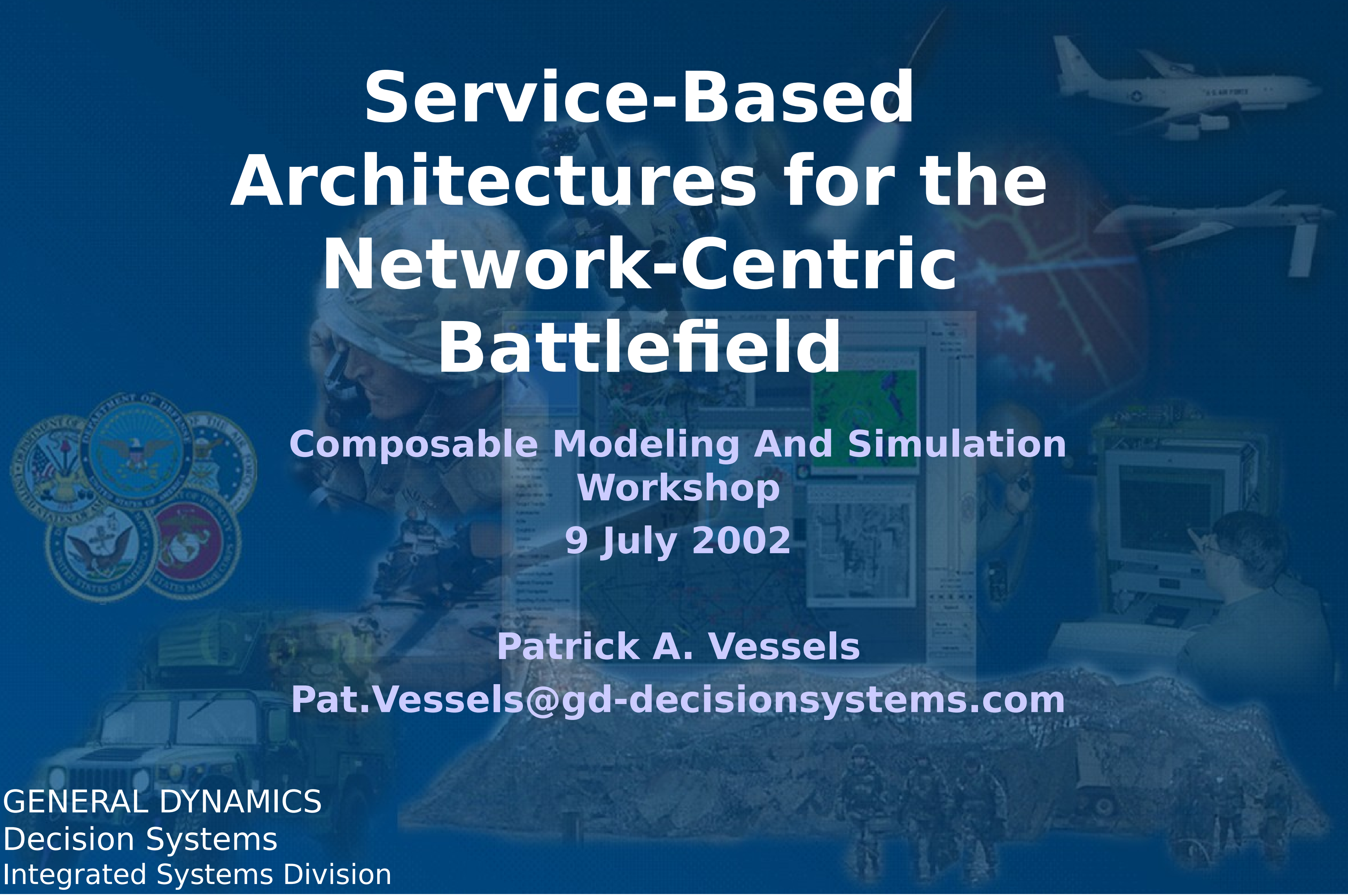


Service-Based Architectures for the Network-Centric Battlefield

Composable Modeling And Simulation Workshop

9 July 2002

Patrick A. Vessels

Pat.Vessels@gd-decisionsystems.com

GENERAL DYNAMICS
Decision Systems
Integrated Systems Division

Introduction to Service-Based Architectures

Architectural Evolution

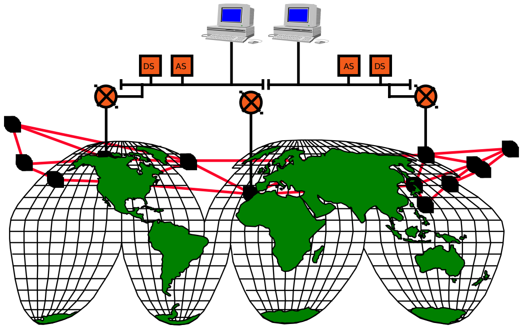
Capabilities
Accessible only
On the Stovepipe
System

Yesterday

Monolithic Vertical Stovepipe Systems

Capabilities
Available only to
Systems that have
Been integrated together

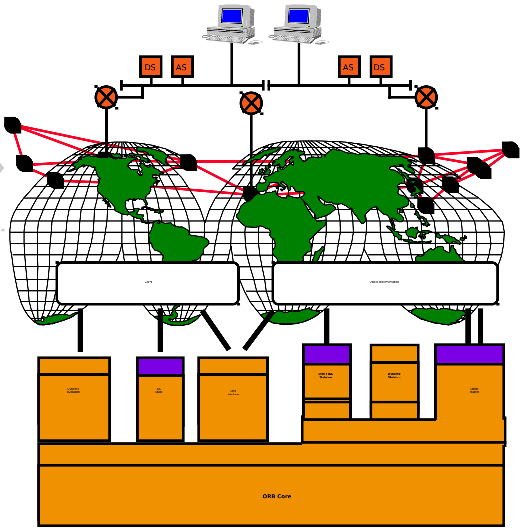
Today



Client/Server
(Predominantly Procedural & RDBMS Based)
Platform Specific

Capabilities
Available anywhere
On the network, even
On non-integrated
systems

Tomorrow

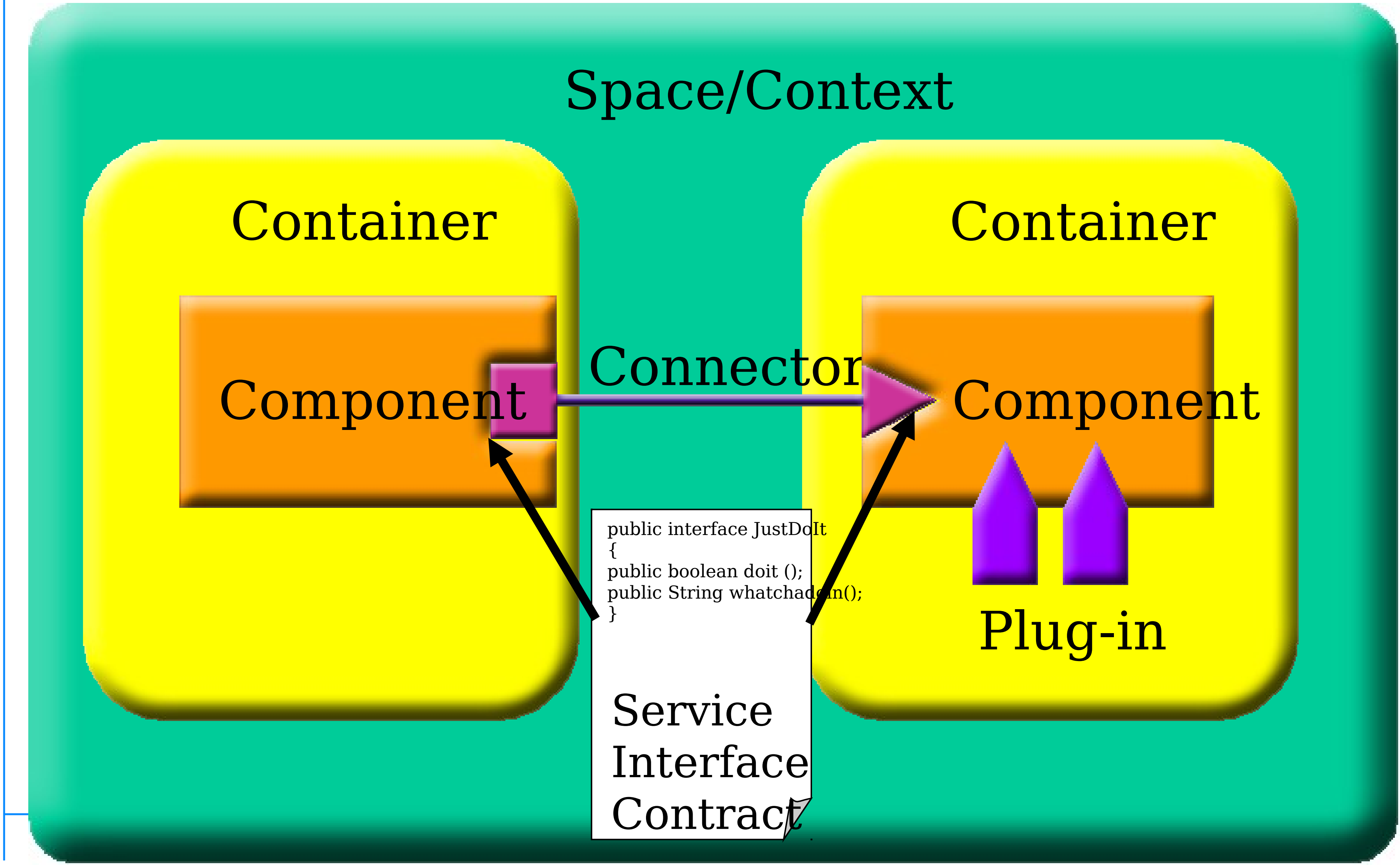


Ad-Hoc Networks
Service-Based
Peer-to-Peer
Platform Independent

What is a Service?

Definition: A **service** is a defined behavior that can be provided by a **component** for use by any **component**, solely based on the defined **Interface**

Key Concepts and Definitions



```
public interface JustDoIt
{
public boolean doit ();
public String whatchadoin();
}
```

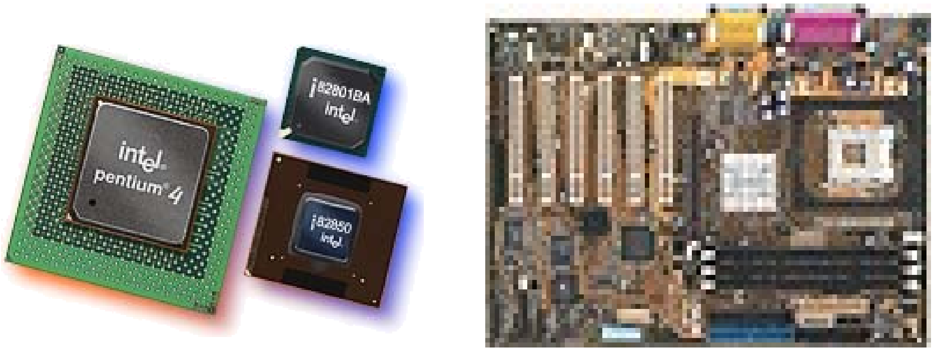
Key Distributed Software Technologies

Technology	Value
HTTP/HTML	3-Tier model for information delivery. The key technology that made the Internet successful
XML	Markup language for standardizing data. Key technology for future web services
Java	Secure, platform independent mechanism for delivering application logic across the network
Jini	Specifically designed to enable discovery of components across the network
JXTA	Peer-to-Peer Discovery & Communications Technology
Web Services	XML Based protocols for information exchange (mostly B2B & E-Commerce)
Openwings	Provides a service model for ad-hoc, distributed systems

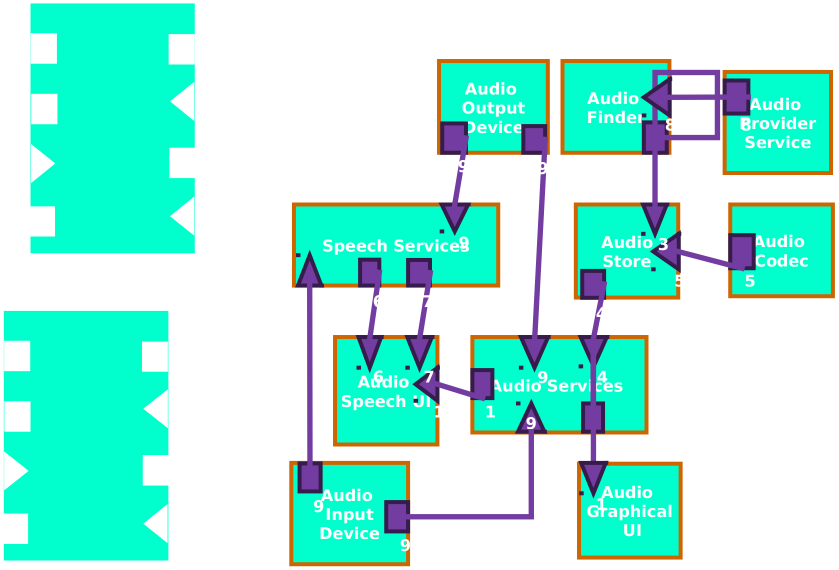
Think Reusable Components

- **Move from developing custom components to integration of components**

Hardware Components



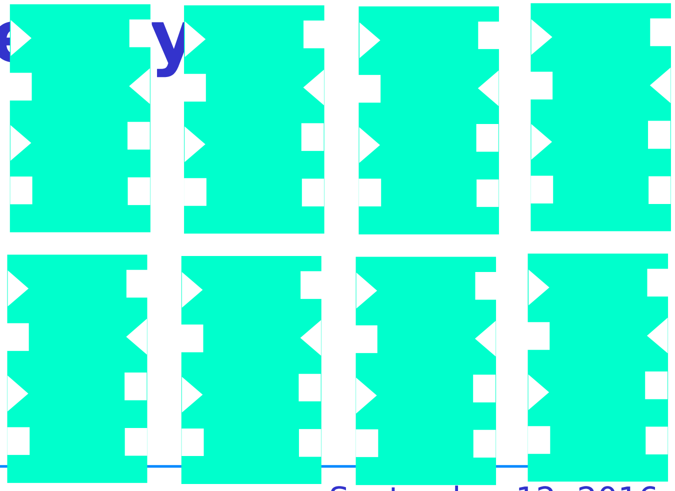
Software Components



Service-Based Design considerations

- **Components may be deployed on different platforms**
  - The Platform / OS of choice changes every three years
- **Components may be deployed in different systems**
  - Components must outlive the systems they are deployed into.
- **Package elements independently**
  - UIs
  - Executable Components
  - Interfaces / Connectors

Component Library

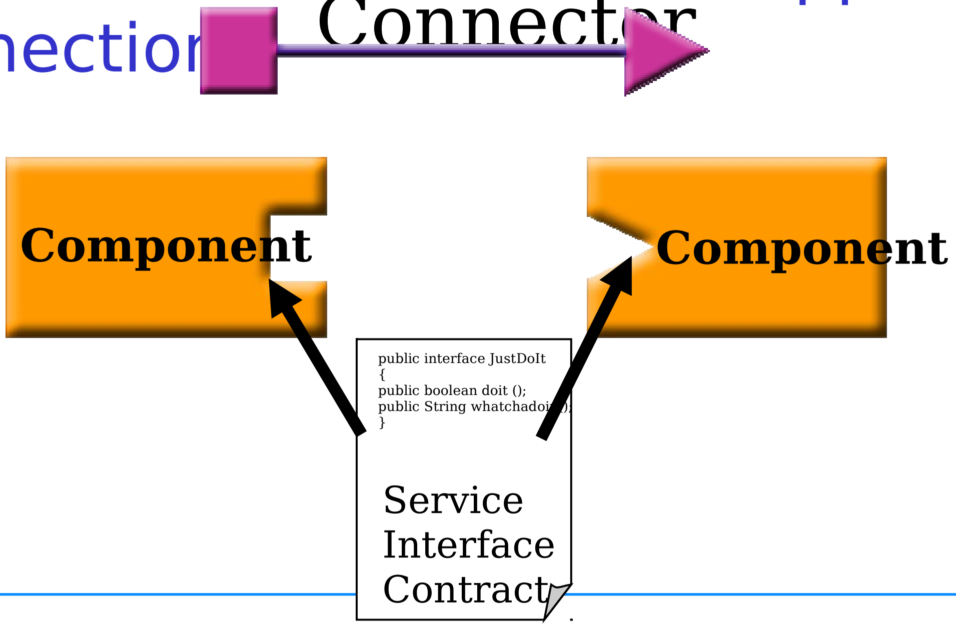


Interfaces are the key to Interoperability

- The Language of the Contract == Java Interface
- An interface allows you to say what a service provides without saying how the service is implemented
- The interface code and comments prescribe the syntax and semantics of the contract.

Interoperability Guideline #1

- **Agree on the interface, not on the protocol**
  - Connectors provide the middleware implementation at runtime – No hard coded protocols
  - Interchangeable
  - Provides a mechanism to support legacy connection



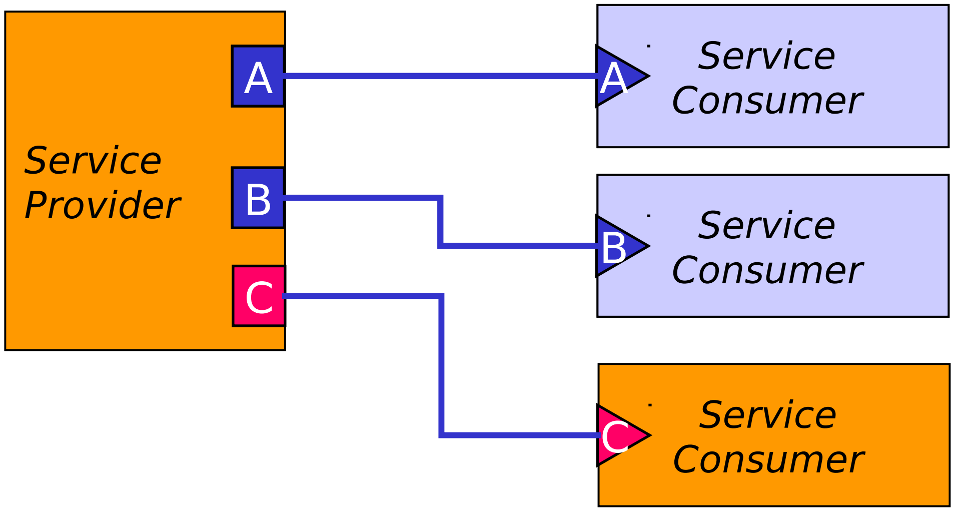
```
public interface JustDoIt
{
public boolean doit ();
public String whatchadoit();
}
```

Interoperability Guideline #2

- **Continue to Support Legacy Interfaces**

Example: a service providing an angle measurement
service A – in degrees (float)
service B – in degrees (double)
service C – in radians (double)

Interface C is the current version, Interface A&B are previous/legacy versions



Service Provider

A

B

C

Service Consumer

Service Consumer

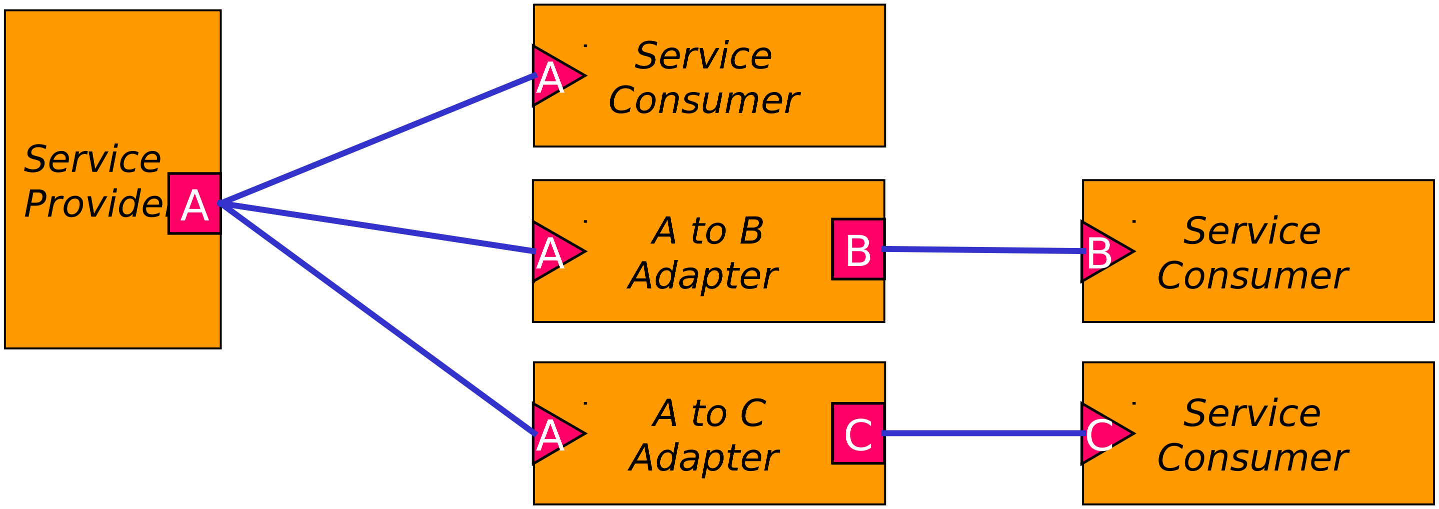
Service Consumer

Legacy Interfac e

Legacy Interface

Interoperability #3

- **Provide Adapters for Legacy Services**
  - Chain connectors together to create adapters



Interoperability Guideline #4

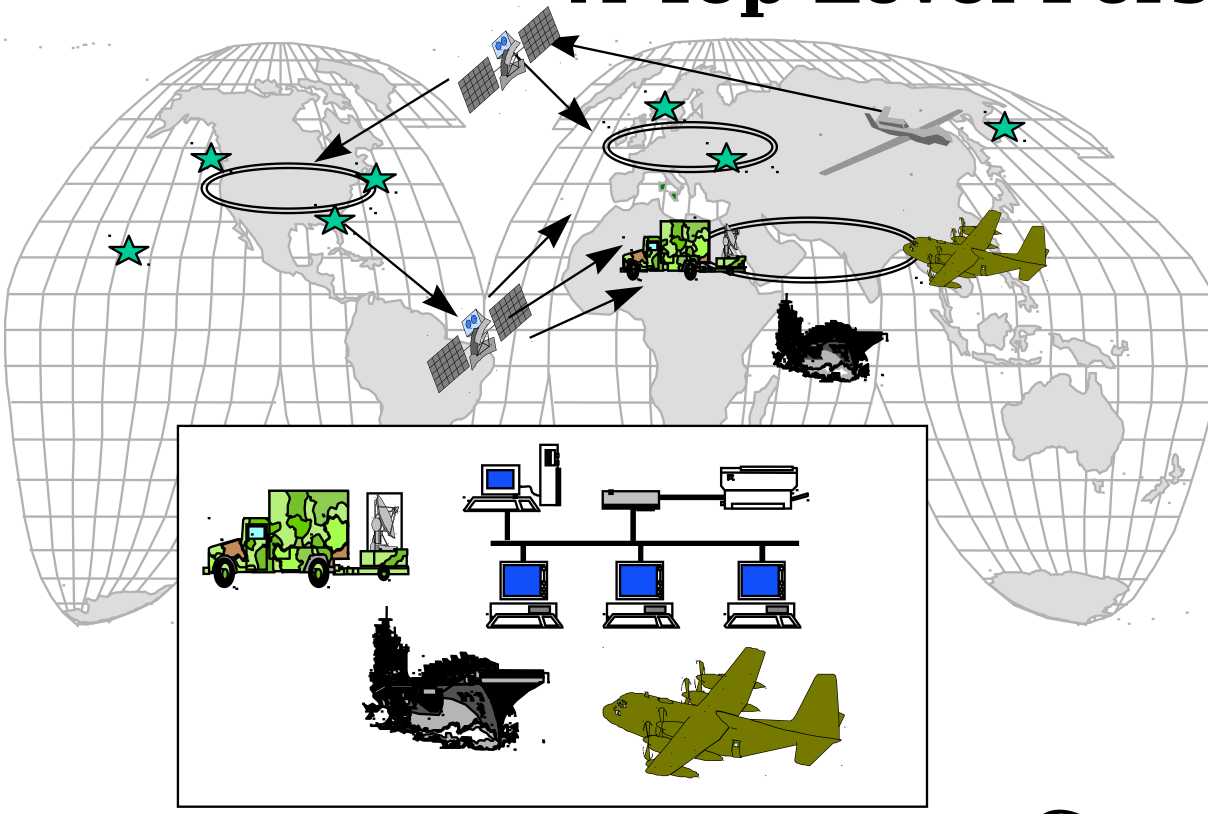
- **Provide the User Interface**
  - ServiceUIs – GUI for a service on a remote machine
  - Applications – installed or dynamically downloaded
  - Applets – dynamically downloaded

OSD Vision for Network-Centric Warfare

OSD Vision for the Future

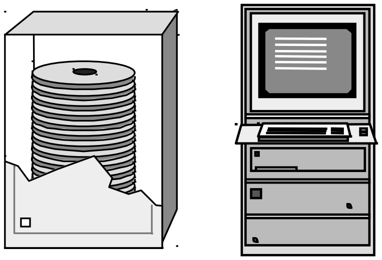
DCGS Architecture

A Top Level Perspective



Global Perspective

A Network Centric Distrib
Multi-INT support Syste



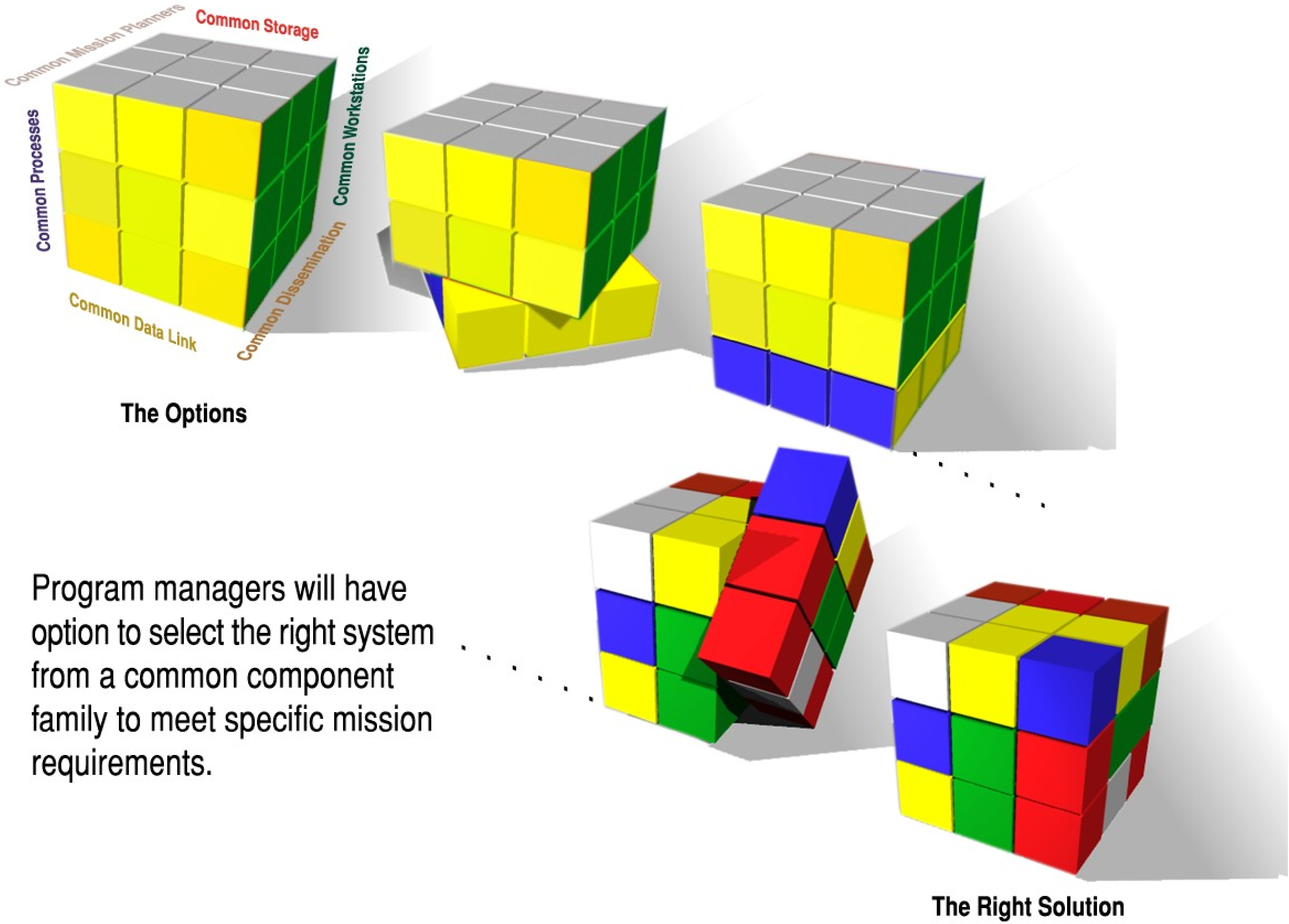
Element Perspective

A Deployable, Modular, Scaleable
Combat System

Component Perspective

Parts of the Elements
Basic architectural modules

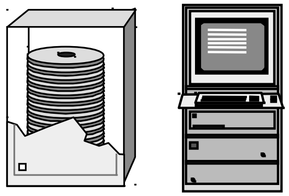
Total Flexibility



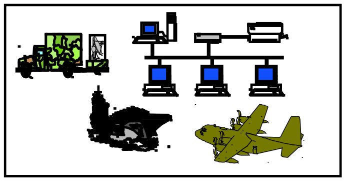
Program managers will have option to select the right system from a common component family to meet specific mission requirements.

SBA Supports…

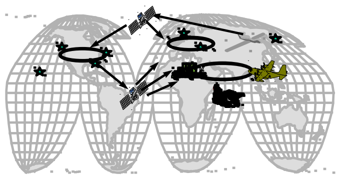
- **Component Perspective**
  - Plug-n-Play hardware & software



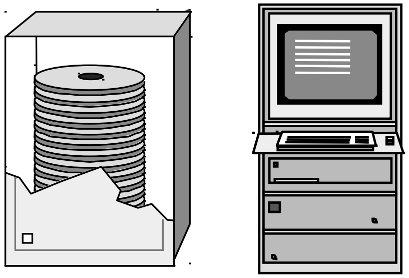
- **Element Perspective**
  - Increased interoperability in a system of systems environment
  - Advanced System-of-Systems concepts



- **Global Perspective**
  - Global access to services via network



SBA support for the Component Perspective



Component Perspective

Parts of the Elements

Basic architectural modules

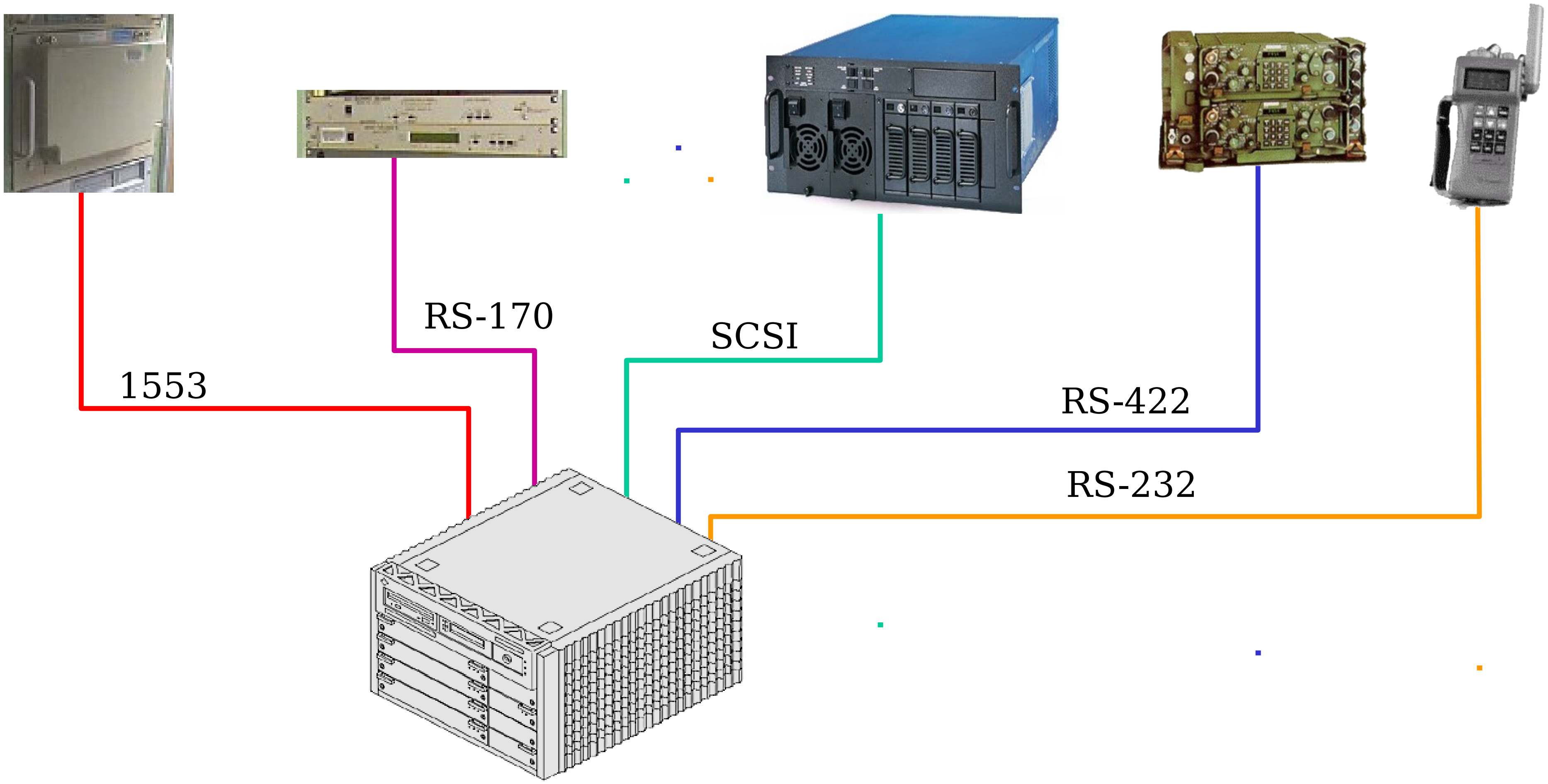
Field-Reconfigurable Systems

Goal: Rapidly reconfigure systems capabilities with little or no administration required

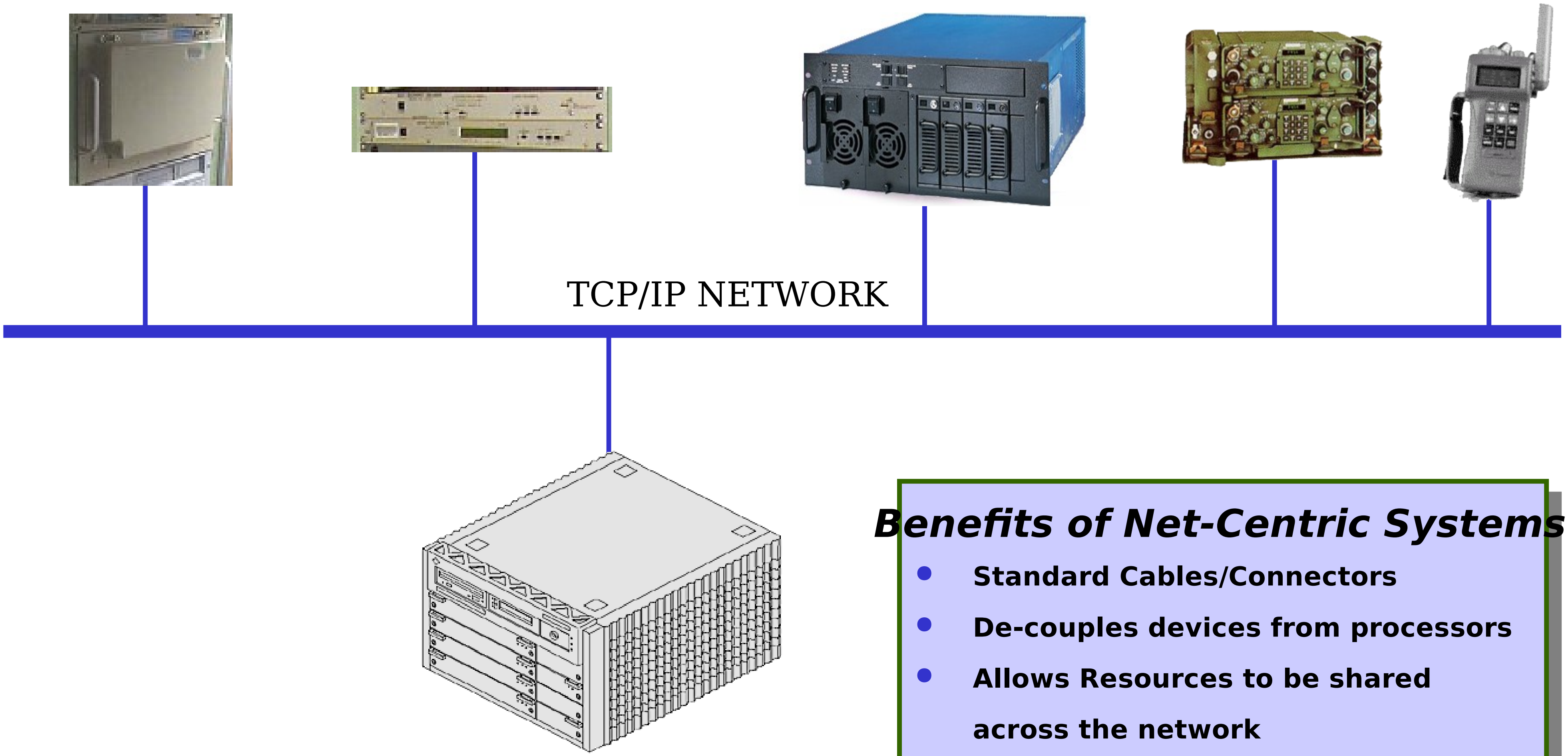
Key Principles

- Hardware & Software are packaged as a unit.
- Hardware has a network interface, processing capability & storage
- Hardware is published as a service on the network

Today’s C4I System Interfaces



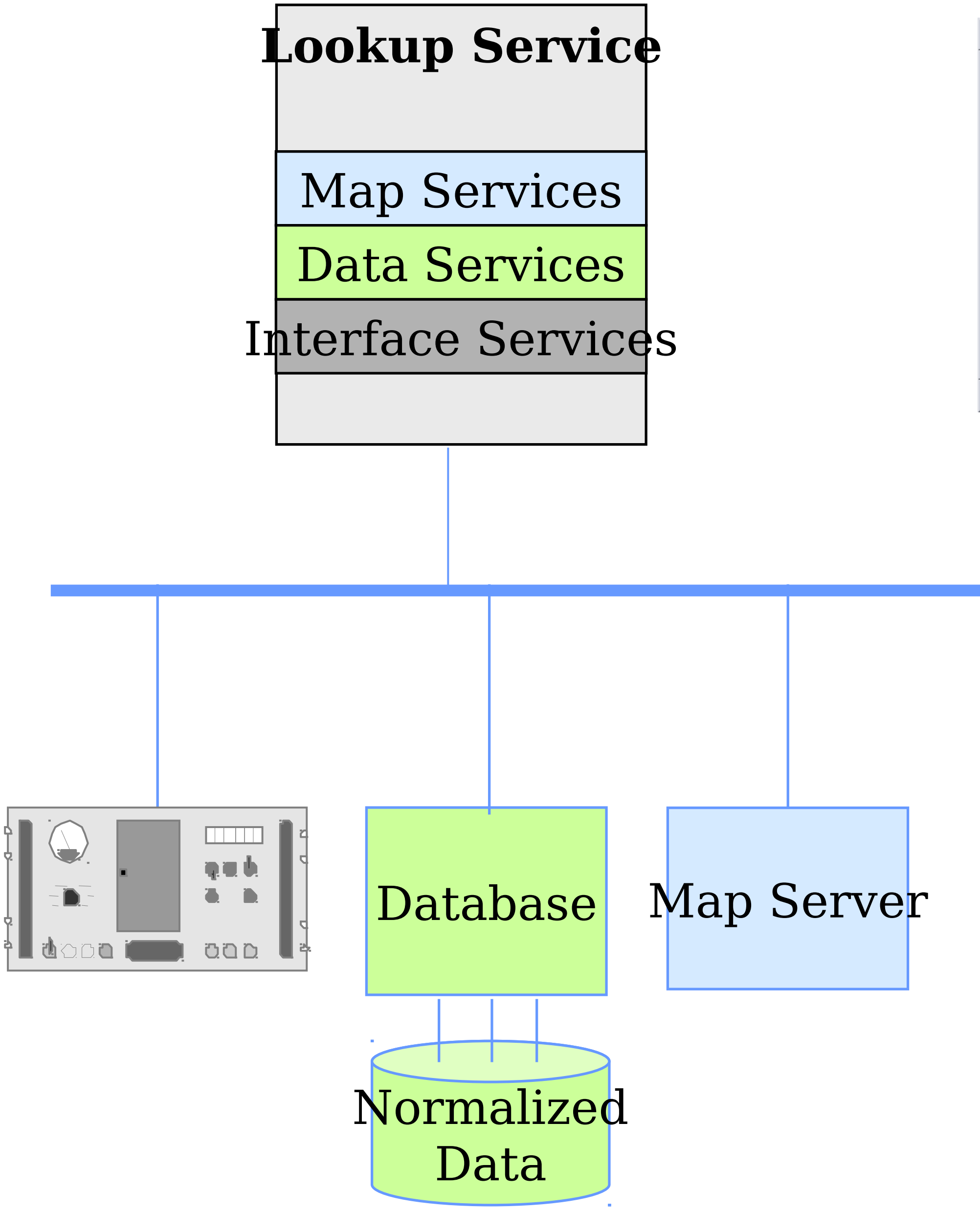
Network Centric Systems

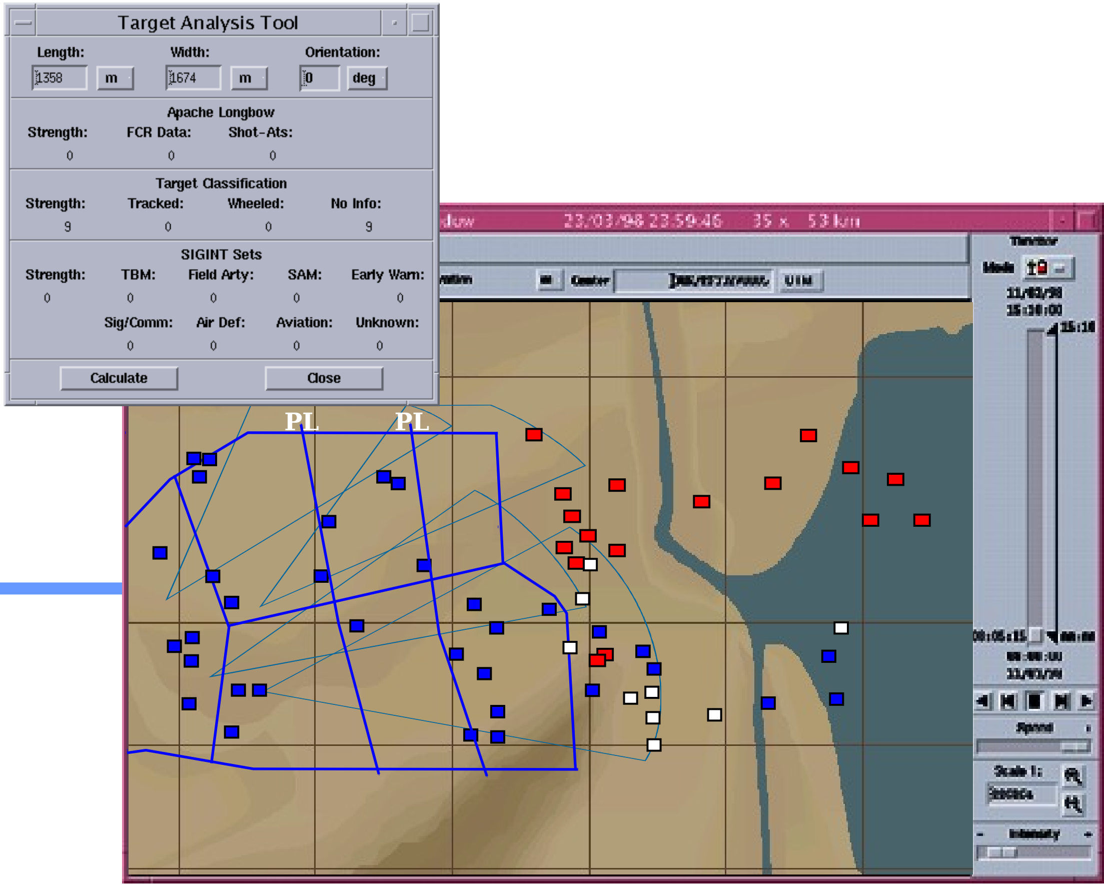


Benefits of Net-Centric Systems

- Standard Cables/Connectors
- De-couples devices from processors
- Allows Resources to be shared across the network
- Easily allows remote monitoring & control

Ad-hoc Component Integration





Target Analysis Tool

Length: 1358 m

Width: 1674 m

Orientation: 0 deg

Apache Longbow

Strength:	FCR Data:	Shot-Ats:
0	0	0

Target Classification

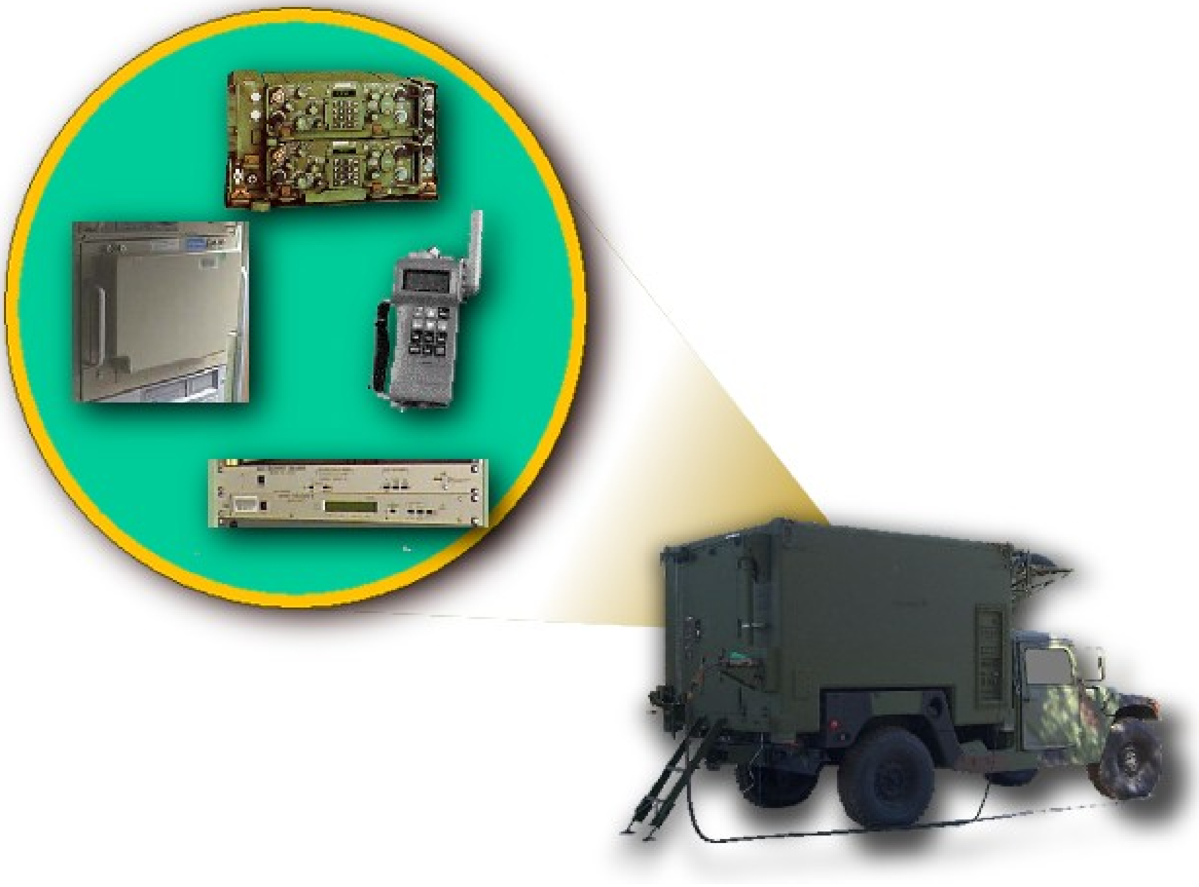
Strength:	Tracked:	Wheeled:	No Info:
9	0	0	9

SIGINT Sets

Strength:	TBM:	Field Arty:	SAM:	Early Warn:
0	0	0	0	0
	Sig/Comm:	Air Def:	Aviation:	Unknown:
	0	0	0	0

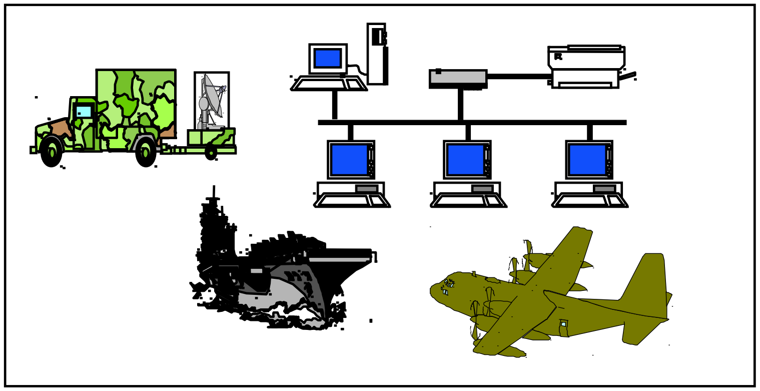
Modular/Scaleable HW/SW Systems

Reconfigurable Based on the Mission



It’s Not All or Nothing. Take Only What You Need

SBA enables OSD’s DCGS Element Perspective



Element Perspective

A Deployable, Modular, Scaleal
Combat System

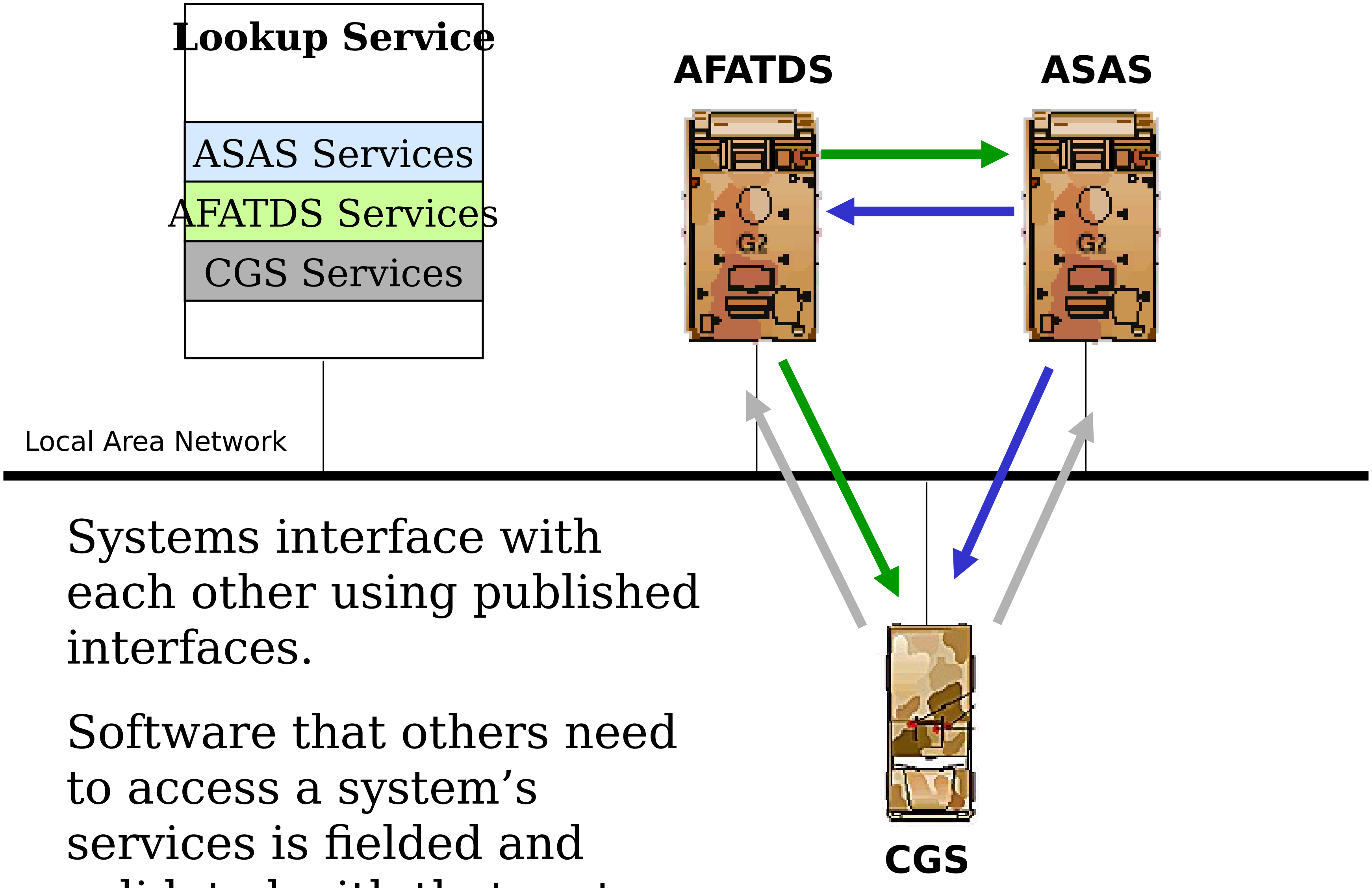
System-of-System Interoperability

Goal: Increase the interoperability between systems by simplifying the interface requirements

Key Principles

- Systems dynamically provide their services to consumers on the network
- Standardized service definitions
- Services can be provided locally or globally

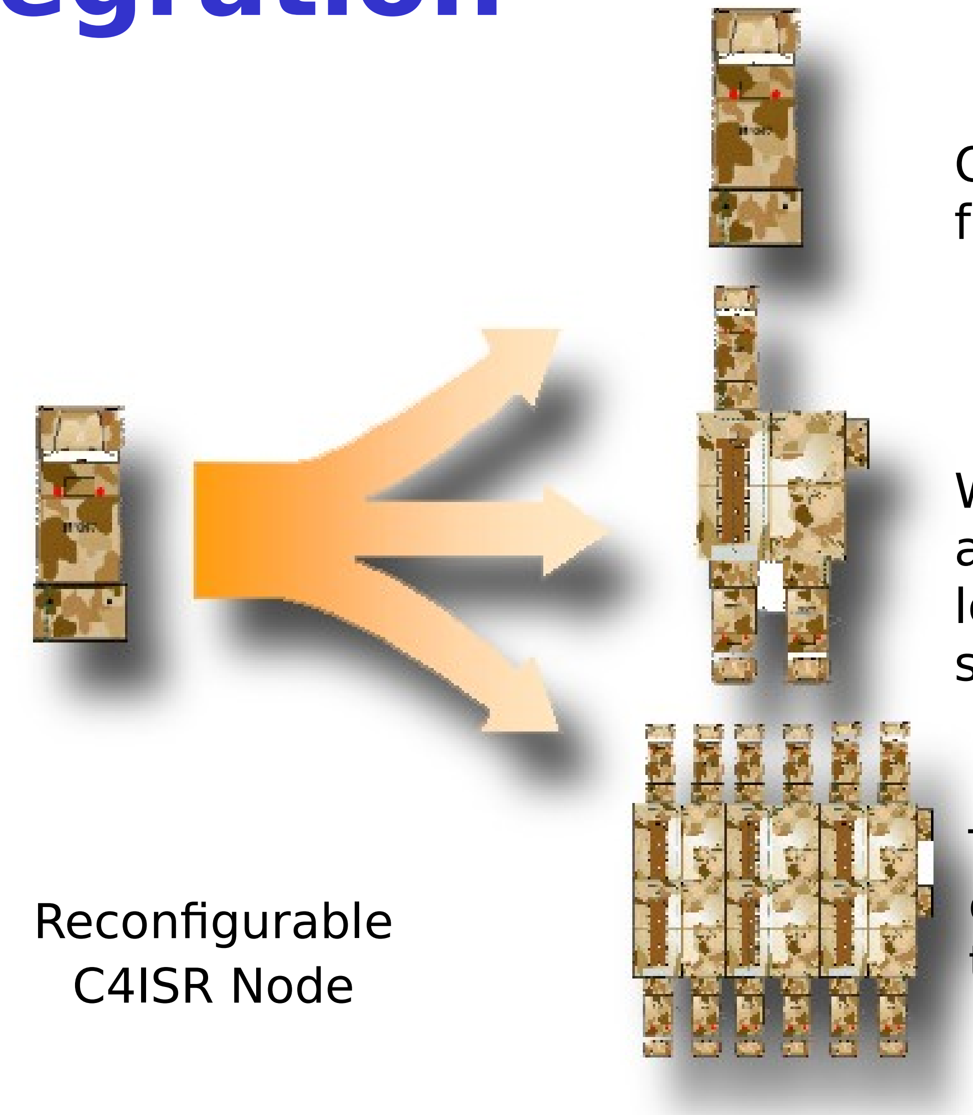
System of Systems Interoperability



Systems interface with each other using published interfaces.

Software that others need to access a system’s services is fielded and validated with that system.

System of System Integration



Operates Standalone for given mission

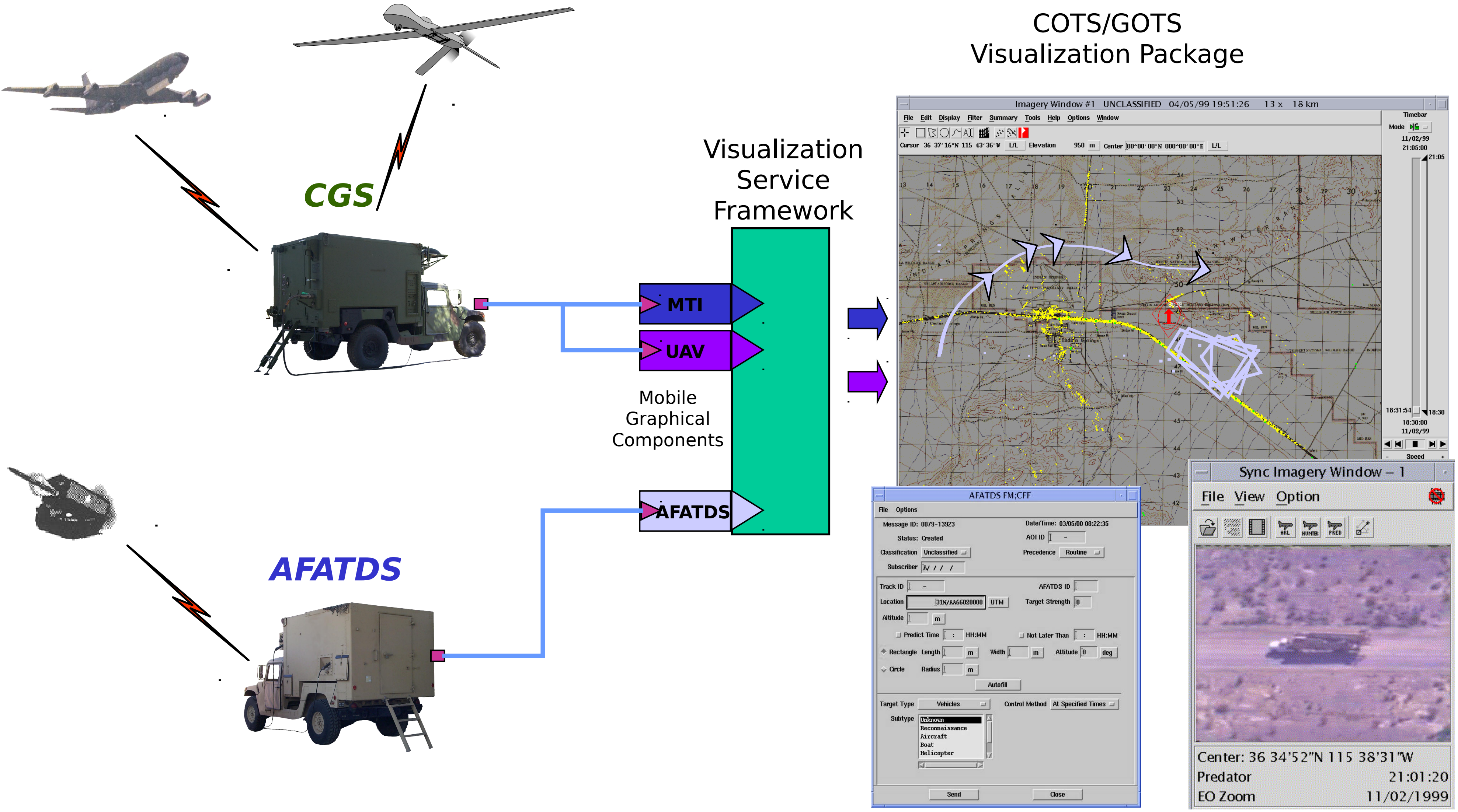
When 2 or more nodes are connected they logically become one system

The more elements, the greater the functionality of the system

Reconfigurable C4ISR Node

Allows sharing of all system resources not just data!

Distributed Services


CGS
AFATDS
MTI
UAV
AFATDS
Visualization Service Framework
Mobile Graphical Components
COTS/GOTS Visualization Package

Imagery Window #1 UNCLASSIFIED 04/05/99 19:51:26 13 x 18 km

File Edit Display Filter Summary Tools Help Options Window

Cursor 36 37′16"N 115 43′36"W L/L Elevation 950 m Center 00^00′00"N 000^00′00"E L/L

Timebar Mode 11/02/99 21:05:00 21:05 18:31:54 18:30 18:30:00 11/02/99 Speed

AFATDS FM;CFF

Message ID: 0079-13923

Date/Time: 03/05/00 08:22:35

Status: Created

AOI ID -

Classification Unclassified

Precedence Routine

Subscriber A/ / / /

Track ID -

AFATDS ID

Location 31N/AA66020000 UTM

Target Strength 0

Altitude m

□ Predict Time : HH:MM

□ Not Later Than : HH:MM

● Rectangle Length m Width m Attitude 0 deg

○ Circle Radius m

Target Type Vehicles

Control Method At Specified Times

Subtype Unknown Reconnaissance Aircraft Boat Helicopter

Sync Imagery Window – 1

Center: 36 34′52"N 115 38′31"W

Predator 21:01:20

EO Zoom 11/02/1999

Targeting Interface Provided by AFATDS

UAV Viewing App Provided by CGS

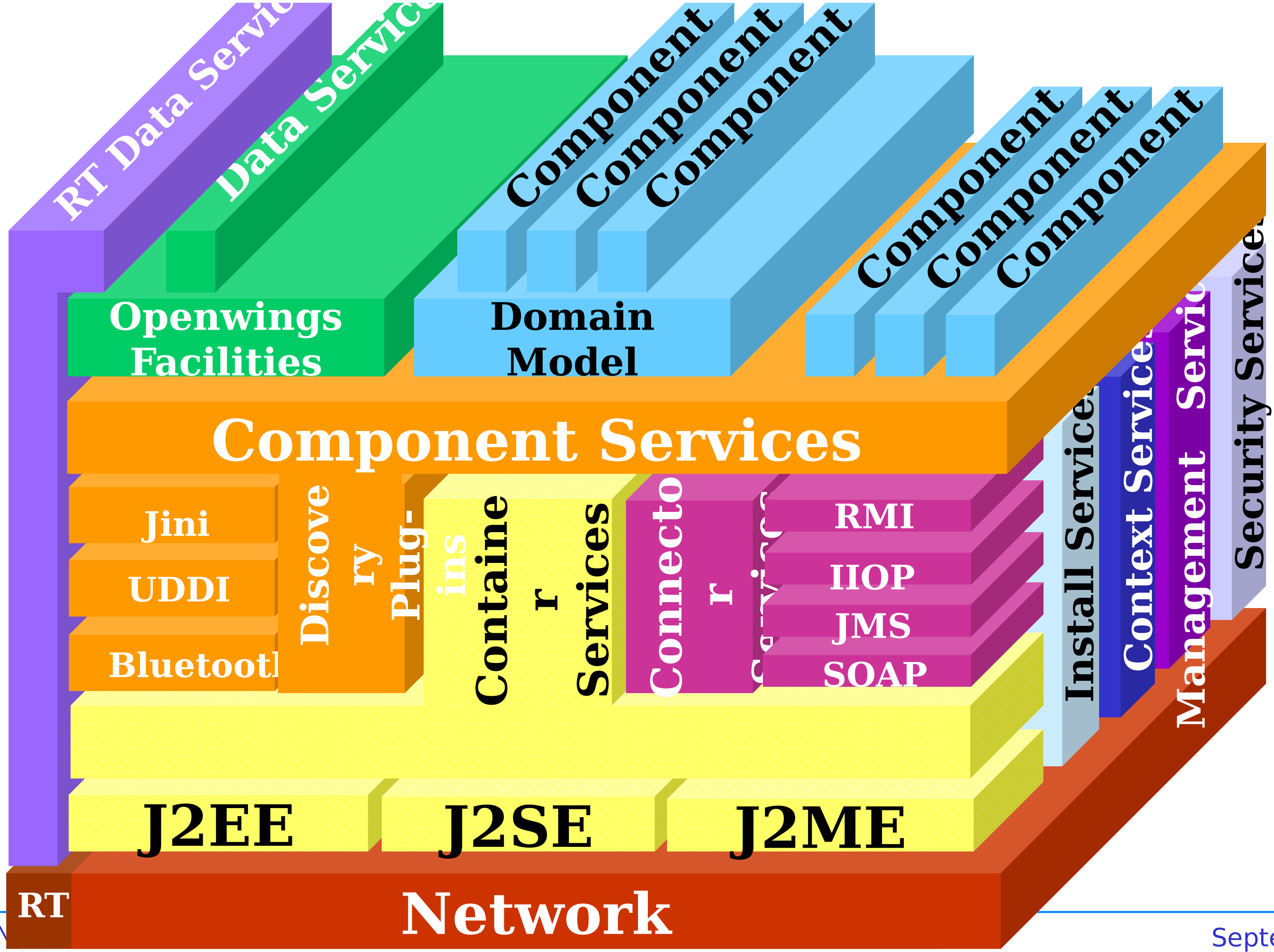
Openwings Overview

What is Openwings?

- Openwings is a consortium for the development of a non-proprietary, open systems framework which enables the development of highly available, secure, distributed systems for mission critical applications

- Focus is on ad-hoc integration of software and hardware components as well as increasing the interoperability in a Systems of Systems environment

Openwings Architecture Model



SBA Challenges

- **Definition of Standard Services**
  - ➤ Standards must be defined for true interoperability
- **Peer to Peer Security**
  - ➤ Mobile code, distributed authentication/authorization, embedded encryption & connection-based security must be addressed
- **Maturing of Key Technologies**
  - JXTA, Web Services and Openwings are some of the most promising

Conclusion

- Client-Server architectures in place today are neither scaleable nor flexible enough to meet the demands of Network Centric Warfare
- Benefits of a Service-Based Architecture are Compelling: Lower development costs, greater flexibility, reduced footprints, greater interoperability
- Core technologies are becoming available, but much work is needed to adapt them to the military domain